APT Hunter:

“Enabling the hunt for abnormalities”

HAO WANG

JOSHUA THEIMER

Introduction

Hao Wang:

- Senior in the EY's cybersecurity advisory practice.
- 5 years of experience in Attack & Penetration (A&P) and Incident Response (IR)

Joshua Theimer:

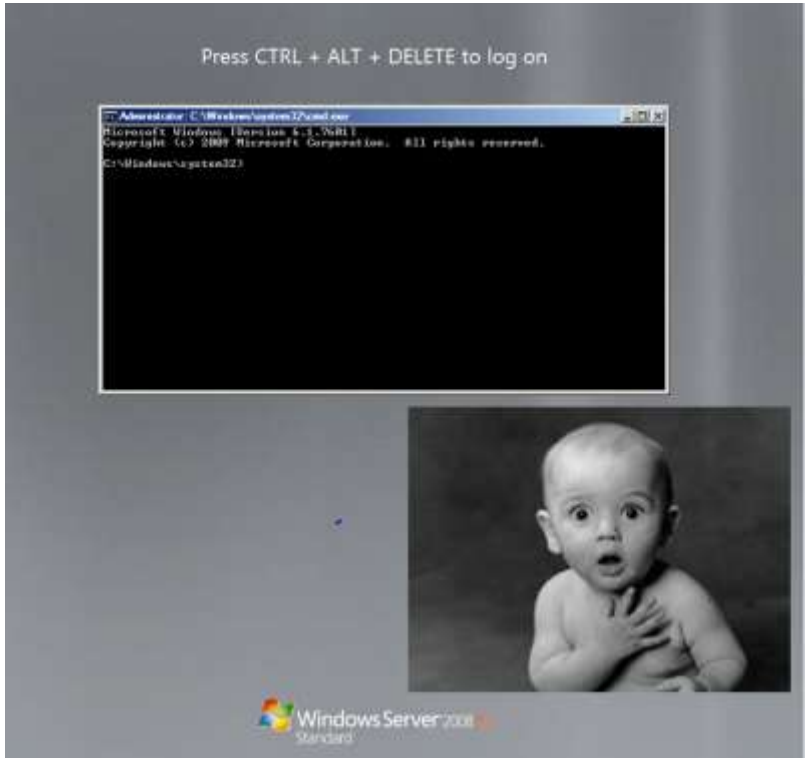
- Manager in the EY's cybersecurity advisory practice; primarily focused on Attack & Penetration (A&P) and Incident Response (IR)

DISCLAIMER: None of the ideas, content, or opinions expressed in this presentation are shared, supported, or endorsed in any manner by our employer.

Motivation

"Know Abnormal…Find Evil"
- SANS poster 2014

Quickly find "abnormal" during our standard A&P testing?



Project Objectives

Goals:

- Independent - Obtain critical information from system and memory to uncover “abnormal activities” within a Windows environment ***without reliance*** on commercial tools or agents; no agent or installation needed
- Scalable - able to ***quickly*** tackle large enterprise-scale hunts
- Lightweight - Pose ***minimal impact*** to analyzed systems

Analysis Requirements:

- Basic IR knowledge required to understand and analyze some of the output via searching & stacking
- Effectiveness increases when parsing fed into an analysis platform

So, how does it work?

1. Establish privileged Windows Management Instrumentation (WMI) connections to remote Windows systems. Use VBScript and WMI queries to extract host level information:
   - File system
   - Memory
   - Network connections
2. Look for "**abnormalities**" from an attack/breach:
   - Enumeration and lateral movement
   - Privilege escalation
   - Persistence mechanisms

Technical Requirements:

1. Privileged Active Directory account (administrator permissions needed on hosts where it will run)
2. VBScript – Windows Management Instrumentation (WMI)
3. .NET Framework 4.6

Current tool capabilities

Enumeration and lateral movement

- Network connections
- System root directory listing
- Local host DNS
- Local administrator group membership
- System information

Privilege escalation

- ShimCache
- AmCache
- WDigest downgrade
- RecentFileCache
- Prefetch

Persistence mechanisms

- Sticky key backdoors
- Rogue services & processes
- Autoruns info
- Scheduled tasks
- “psexesvc.exe”
- Uncommon RDP ports

Subset of capabilities discussed today

Enumeration and lateral movement

- ~~Network connections~~
- ~~System root directory listing~~
- ~~Local host DNS~~
- ~~Local administrator group membership~~
- ~~System information~~

Privilege escalation

- **ShimCache**
- **AmCache**
- **WDigest downgrade**
- **RecentFileCache**
- ~~Prefetch~~

Persistence mechanisms

- **Sticky key backdoors**
- **Rogue services & processes**
- **Autoruns info**
- **Scheduled tasks**
- **WMI Persistence**
- ~~"psexesvc.exe"~~
- ~~Uncommon RDP ports~~

Abnormal File Execution

ShimCache, AmCache, RecentFileCache

- Identification of suspicious file names (i.e. *dump*, *hash*, *password*, single character)
- Identify the use of utilities preferred by attackers (i.e. at.exe, rar.exe, psexec.exe, psexesvc.exe, wmic.exe, powershell.exe, cscript.exe, wscript.exe, mofcomp.exe, scrcons.exe, csc.exe w/ installutil.exe)
- Identify binaries run from suspicious paths (i.e. c:\temp, c:\wmpub, c:\Windows\addins, C:\users, C:\PerfLogs)
- Identify known malware based on filename and file size
- Very powerful for large-scale search & stacking analysis

ShimCache

What is ShimCache?

- Windows registry keys storing file metadata of executed files
- Used by Microsoft for lookups to identify application compatibility issues; used to determine if modules require shimming for compatibility

What information can we get from ShimCache?

- file path, size, last modified time, and last execution time (depending on OS)
- files executed or created on the file system

Location of ShimCache?

- HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatibility [or] AppCompatCache

References:

- https://dl.mandiant.com/EE/library/Whitepaper_ShimCacheParser.pdf
- https://github.com/mandiant/ShimCacheParser
- http://binaryforay.blogspot.com/2015/05/introducing-appcompatcacheparser.html
- www.woanware.co.uk/forensics/shimcacheparser.html
- Hunting in the Dark HTCIA 2015 by Ryan Kazanciyan

ShimCache (cont.)

ShimCache Example #1: After script run, look for evidence of PsExec ("PSEXESVC.EXE") execution in tool output:

HostName	Last Modified	Last Update	Path	File Size	Exec Flag
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\ipconfig.exe	61440	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\userinit.exe	24064	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\taskkill.exe	81408	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\net1.exe	120320	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\net.exe	40960	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\reg.exe	78336	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\System32\logon.scr	508928	N/A
192.168.100.28	3/27/2014 14:44	N/A	C:\myftp2\myftp2.pdf	276994	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\sethc.exe	31744	N/A
192.168.100.28	8/6/2013 23:08	N/A	C:\wce32p.exe	208384	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\netstat.exe	31744	N/A
192.168.100.28	3/19/2014 14:39	N/A	C:\WINDOWS\wce-u.exe	466944	N/A
192.168.100.28	3/27/2014 17:02	N/A	C:\WINDOWS\PSEXESVC.exe	189792	N/A
192.168.100.28	11/30/1979 5:00	N/A	C:\myftp2\myftp2.exe	300544	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\at.exe	24576	N/A
192.168.100.28	3/7/2014 4:53	N/A	C:\PsExec.exe	396480	N/A

ShimCache (cont.)

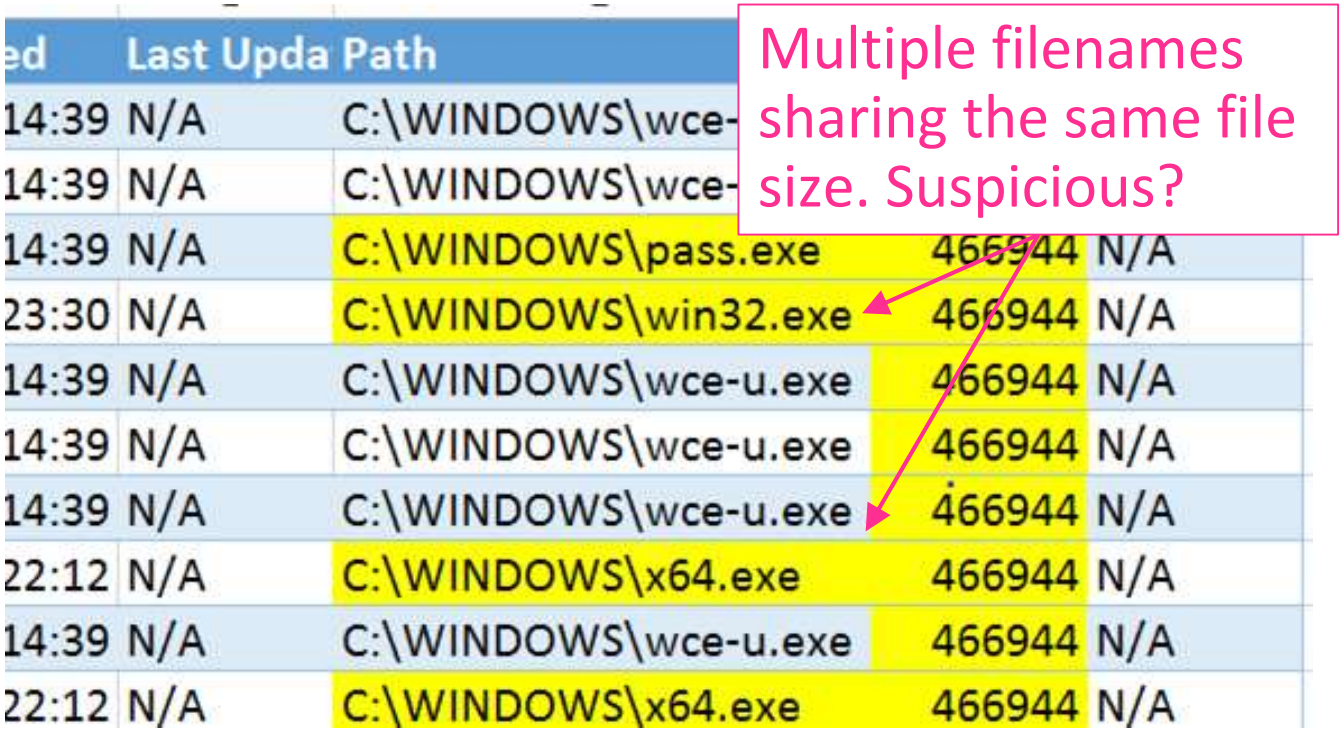
ShimCache Example #1: After script run, look for evidence of PsExec ("PSEXESVC.EXE") execution in tool output:

HostName	Last Modified	Last Update	Path	File Size	Exec Flag
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\ipconfig.exe		
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\userinit.exe		
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\taskkill.exe		
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\net1.exe		
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\net.exe		
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\reg.exe		
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\System32\logon.scr	508928	N/A
192.168.100.28	3/27/2014 14:44	N/A	C:\myftp2\myftp2.pdf	276994	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\sethc.exe	31744	N/A
192.168.100.28	8/6/2013 23:08	N/A	C:\wce32p.exe	208384	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\netstat.exe	31744	N/A
192.168.100.28	3/19/2014 14:39	N/A	**C:\WINDOWS\wce-u.exe**	466944	N/A
192.168.100.28	3/27/2014 17:02	N/A	C:\WINDOWS\PSEXESVC.exe	189792	N/A
192.168.100.28	11/30/1979 5:00	N/A	C:\myftp2\myftp2.exe	300544	N/A
192.168.100.28	3/25/2003 12:00	N/A	C:\WINDOWS\system32\at.exe	24576	N/A

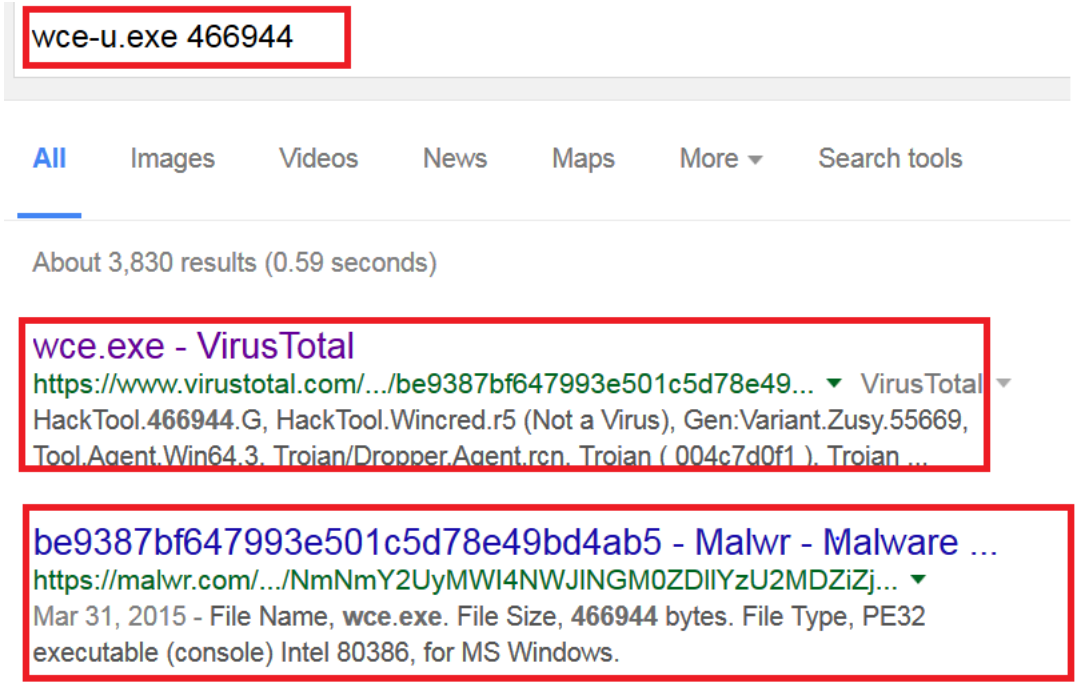
Look at adjacent ShimCache entries for other suspicious files. Does anything look suspicious?

ShimCache (cont.)

ShimCache Example #2: Look for evidence of suspicious entries by filename and file size in tool output:



ed	Last Upda	Path		
14:39	N/A	C:\WINDOWS\wce-		
14:39	N/A	C:\WINDOWS\wce-		
14:39	N/A	C:\WINDOWS\pass.exe	466944	N/A
23:30	N/A	C:\WINDOWS\win32.exe	466944	N/A
14:39	N/A	C:\WINDOWS\wce-u.exe	466944	N/A
14:39	N/A	C:\WINDOWS\wce-u.exe	466944	N/A
14:39	N/A	C:\WINDOWS\wce-u.exe	466944	N/A
22:12	N/A	C:\WINDOWS\x64.exe	466944	N/A
14:39	N/A	C:\WINDOWS\wce-u.exe	466944	N/A
22:12	N/A	C:\WINDOWS\x64.exe	466944	N/A



AmCache

What is AmCache?

- On Windows 7+ and Server 2008+
- Shim database
- Store data for recent run programs / applications

What information can we get from AmCache?

- Useful in identifying recent run files
- Detailed executable information (i.e. file name, full file path, sha1, file timestamps, PE information)

Location of AmCache?

- C:\Windows\AppCompat\Programs\Amcache.hve

References:

- http://www.swiftforensics.com/2013/12/amcachehve-in-windows-8-goldmine-for.html
- https://github.com/williballenthin/python-registry/blob/master/samples/amcache.py
- http://binaryforay.blogspot.com/2015/07/amcacheparser-reducing-noise-finding.html

AmCache

AmCache Example #1: Look for evidence of suspicious process creation in tool output:

A	E	G	H	I	J	L
HostName	VolumeIDLastWriteTimest	FileIDLastWriteTimestamp	SHA1	FullPath	FileExtension	LastModified2
10.10.1.16	3/30/2016 20:25	3/30/2016 20:25	0804008abcd0bbd	C:\inetpub\procdump.exe	.exe	4/29/2014 13:11
10.10.1.16	3/30/2016 20:25	3/30/2016 20:25	8f7e3201b9c485b8	C:\inetpub\procdump64.exe	.exe	3/30/2016 20:25
10.10.1.16	3/30/2016 20:25	3/30/2016 20:17	0c5a8a0c11b9fcad	C:\Windows\PSEXESVC.exe	.exe	3/30/2016 20:17
10.10.1.7	3/30/2016 20:30	3/30/2016 20:29	a4d99d2e581bb41	C:\inetpub\ddump.exe	.exe	3/10/2014 15:45
10.10.1.7	3/30/2016 20:30	3/30/2016 20:29	0804008abcd0bbd	C:\inetpub\procdump.exe	.exe	4/29/2014 13:11
10.10.1.7	3/30/2016 20:30	3/30/2016 20:30	828fab0c1ed06f36	C:\inetpub\mimikatz.exe	.exe	6/18/2014 0:42
10.10.1.7	3/30/2016 20:30	3/30/2016 20:29	e5a2d507a0b609e5	C:\Windows\Temp\g64-b2c.ex	.exe	3/30/2016 20:29
10.10.1.7	3/30/2016 20:30	3/30/2016 20:29	8f7e3201b9c485b8	C:\inetpub\procdump64.exe	.exe	3/30/2016 20:29
10.10.1.7	3/30/2016 20:30	3/30/2016 20:28	0c5a8a0c11b9fcad	C:\Windows\PSEXESVC.exe	.exe	3/30/2016 20:28
10.10.1.15	3/25/2016 23:08	3/20/2016 12:24	3e525b7e35a87ab1	C:\PerfLogs\RawCopy.exe	.exe	12/2/2015 15:03

RecentFileCache

What is RecentFileCache?

- Windows 7
- Shim database
- Used by ProgramDataUpdater to store data for recent process creation

What information can we get from RecentFileCache?

- Useful in identifying recent process creation
- Limited executable information (i.e. file name and file path)

Location of RecentFileCache?

- C:\Windows\AppCompat\Programs\RecentFileCache.bcf
- Replaced by AmCache in Windows 8+

References:

- http://journeyintoir.blogspot.in/2013/12/revealing-recentfilecachebcf-file.html
- https://github.com/sysforensics/RecentFileCacheParser

RecentFileCache

RecentFileCache Example #1: Look for evidence of suspicious process creation in tool output:

HostName	FilePath
192.168.209.159	c:\windows\system32\atbroker.exe
192.168.209.159	c:\windows\psexesvc.exe
192.168.209.159	c:\perflogs\proccoms.bin
192.168.209.159	c:\perflogs\procdump64.exe
192.168.209.147	c:\windows\psexesvc.exe
192.168.209.147	c:\windows\system32\chcp.com

Sticky Key (backdoors)

What are Sticky Key backdoors?

- Sticky Keys are designed for people who have difficulty holding down two or more keys simultaneously. Attackers can replace the accessibility programs (through file replacement or registry modification with programs that provide SYSTEM-level shell access.(Network Level Authentication will not stop Sticky key backdoors.)

What is the meaning of a Sticky Key backdoor?

- Useful to look for identifying (legacy) compromised hosts
- Often used as a backup tactic to persist on a select number of hosts

Location of Sticky Key backdoors?

- Various locations (Please see following slides.)

References:

- http://carnal0wnage.attackresearch.com/2012/04/privilege-escalation-via-sticky-keys.html
- http://zachgrace.com/2015/03/23/hunting-sticky-keys-backdoors.html
- http://www.crowdstrike.com/blog/registry-analysis-with-crowdresponse/

Sticky Key backdoor locations

1. File replacement

File sethc.exe or utilman.exe replaced with another file, typically cmd.exe or explorer.exe

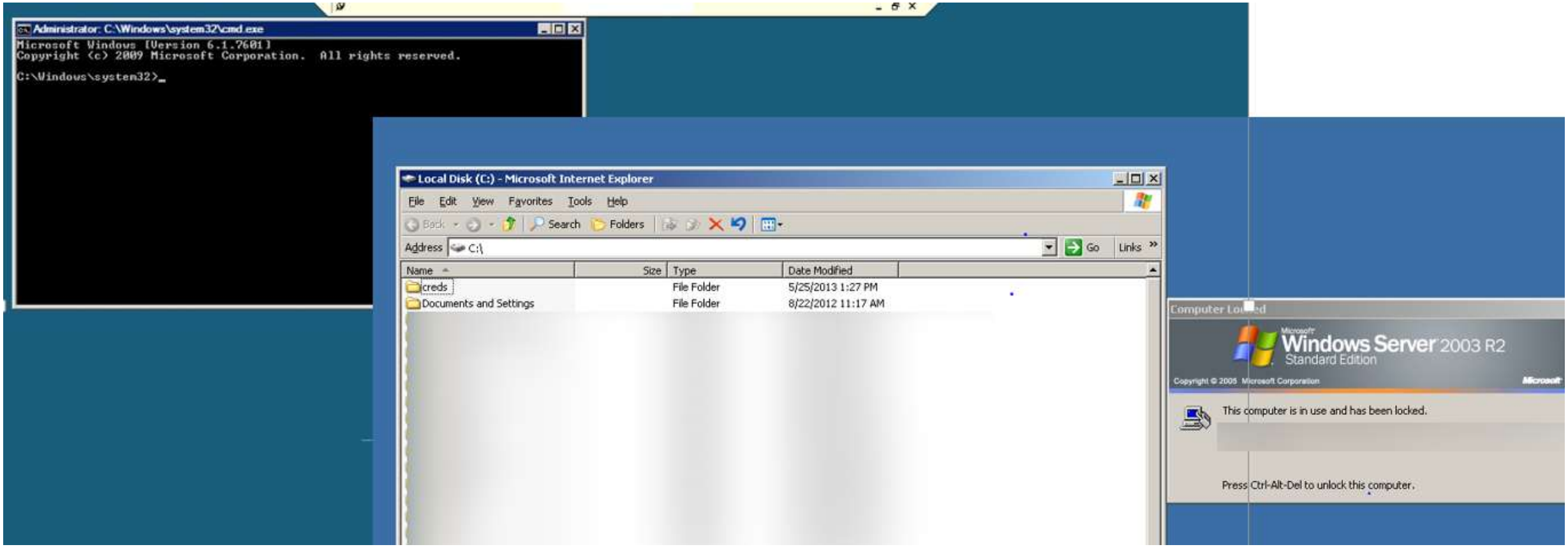
```
Copy C:\Windows\system32\cmd.exe
C:\Windows\system32\sethc.exe /y

Copy C:\Windows\system32\cmd.exe
C:\Windows\system32\utilman.exe /y

Copy C:\Windows\explorer.exe
C:\Windows\system32\utilman.exe /y

Copy C:\Windows\explorer.exe
C:\Windows\system32\sethc.exe /y
```

2. Registry Modification

Make a registry modification to launch a debugger anytime one of the following $FILES is executed

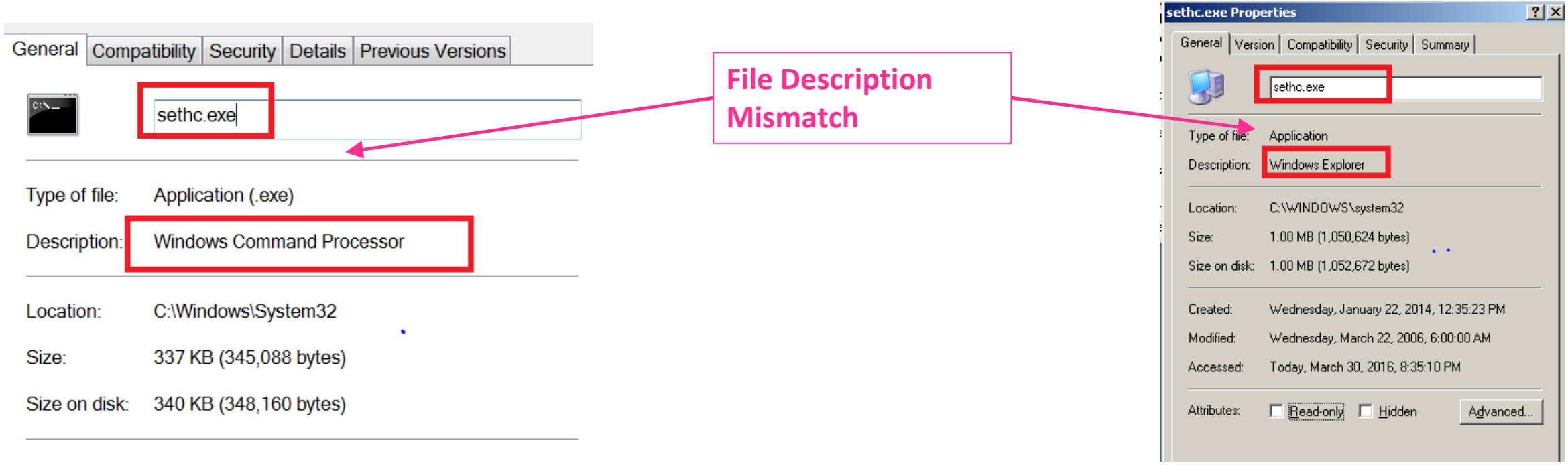
```
REG ADD "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Image File Execution
Options\ $FILES " /v Debugger /t REG_SZ /d
"C:\windows\system32\cmd.exe"

REG ADD "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Image File Execution
Options\ $FILES " /v Debugger /t REG_SZ /d
"C:\windows\system32\explorer.exe"


$FILES = (sethc.exe ,utilman.exe, osk.exe,
narrator.exe, magnify.exe,
displayswitch.exe )
```

Sticky key backdoors



Detection of Sticky key backdoor via 1. File Replacement

Look for evidence of file description mispatch. Do sethc.exe or utilman.exe have the legitimate file descriptions?



Detection of Sticky key backdoor via 2. Registry Modification

Check if the debugger has been set up with certain binaries for the following registries:

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
 Options\setch.exe
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
 Options\utilman.exe
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
 Options\osk.exe
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
 Options\narrator.exe
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
 Options\magnify.exe
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
 Options\displayswitch.exe
```

Sticky Key backdoors

Sticky Key Example #1: Look for evidence of found Sticky Key replacements in tool output:

A	G	H	I	J
HostName	RogueUtilManRegistryCheck	RogueOskRegistryCheck	RogueNarratorRegistryCheck	RogueMagnifyRegistryCheck
10.10.1.16	found:"C:\windows\system32\cmd.exe"	stickykey not found	stickykey not found	stickykey not found
10.10.1.7	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.15	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.6	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.8	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.5	stickykey not found	stickykey not found	found:"C:\windows\system32\cmd.exe"	stickykey not found
10.10.1.10	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.12	stickykey not found	stickykey not found	stickykey not found	stickykey not found

HostName	RogueSethcFileCheck1	RogueUtilManFileCheck1	RogueSethcFileCheck2	RogueUtilManFileCheck2	RogueSethcRegistryCheck
10.10.1.16	stickykey not found	stickykey not found	stickykey not found	stickykey not found	found:"C:\windows\system32\explorer.e:
10.10.1.7	stickykey not found	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.15	stickykey not found	stickykey not found	stickykey not found	stickykey not found	found:"C:\Windows\system32\cmd.exe"
10.10.1.6	stickykey not found	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.8	stickykey not found	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.5	found:Windows Command	stickykey not found	found:Windows Command Process	stickykey not found	stickykey not found
10.10.1.10	stickykey not found	stickykey not found	stickykey not found	stickykey not found	stickykey not found
10.10.1.12	stickykey not found	stickykey not found	stickykey not found	stickykey not found	stickykey not found

WMI Persistence

What is the WMI Persistence?

- Using Windows Management Instrumentation (WMI) to create persistence for malicious payloads through the creation of a permanent WMI event subscription. Payloads can be run with SYSTEM privileges in sneaky ways.

Significance of WMI backdoors?

- Useful for finding compromised hosts missed in prior investigation and eradication
- WMI persistence is relatively hard to detect, there are minimal artifacts on which to trigger alerts

Ways to find WMI persistence?

- Registry key created from “Win32_LocalTime” using WMI EventFilter
- Enumerate / Stacking instances WMI classes:

  EventFilter (Payload triggering condition), EventConsumer (Actual payload), FilterToConsumerBinding (Link between condition and payload)

References:

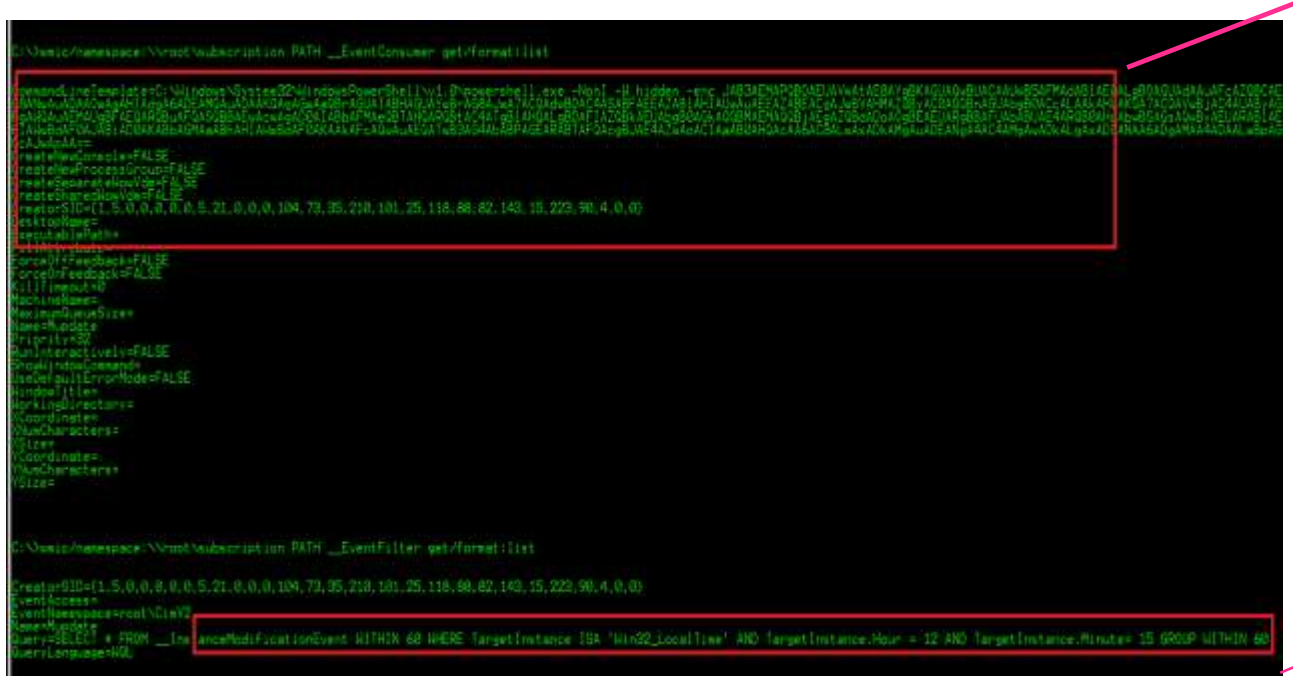
- http://la.trendmicro.com/media/misc/understanding-wmi-malware-research-paper-en.pdf
- https://www.blackhat.com/docs/us-15/materials/us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent%20Asynchronous-And-Fileless-Backdoor-wp.pdf
- https://dl.mandiant.com/EE/library/MIRcon2014/MIRcon_2014_IR_Track_There%27s_Something_About_WMI.pdf
- https://github.com/PowerShellEmpire/Empire https://github.com/PowerShellMafia/PowerSploit
- https://www.secureworks.com/blog/wmi-persistence

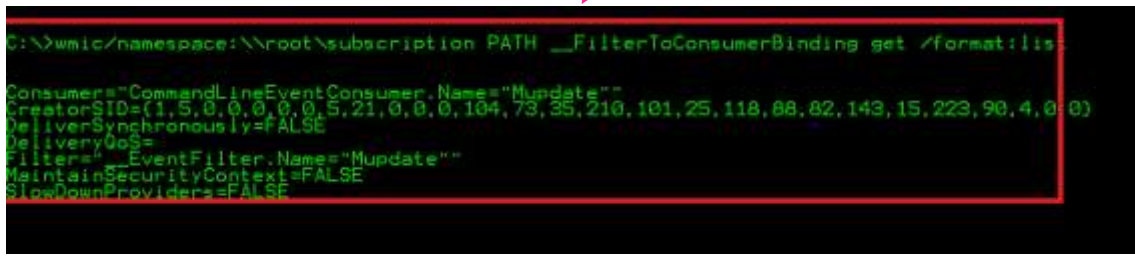
WMI Persistence

WMI Persistence Example #1: Look for suspicious instances from EventFilter, EventConsumer, and FilterToConsumerBinding:

EventConsumer – "malicious payload"

FilterToConsumerBinding – Link for payload and condition

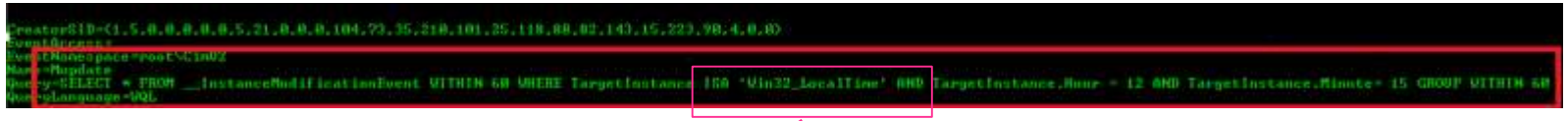




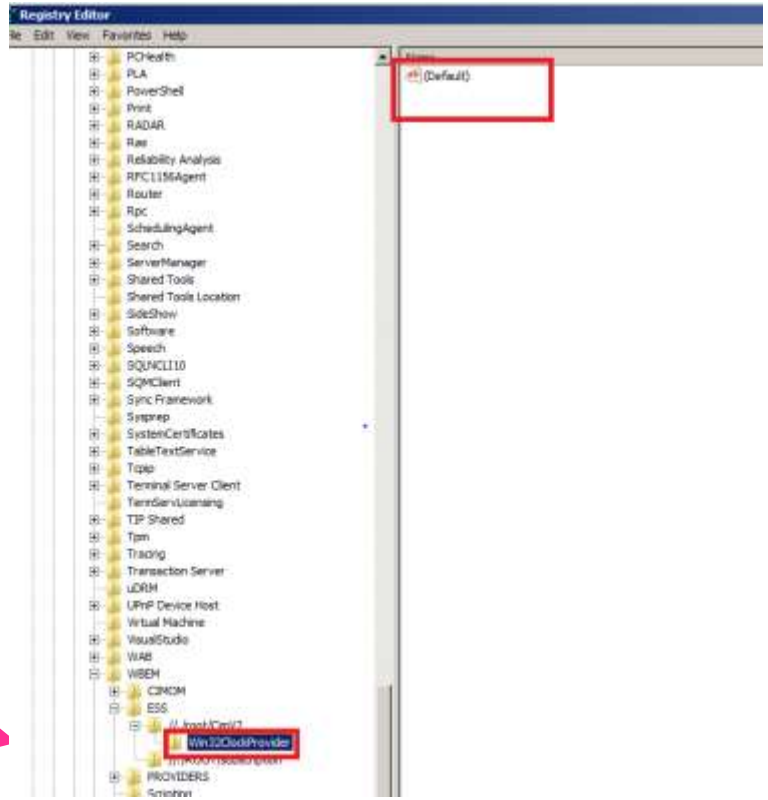
EventFilter – Payload triggering condition via "Win32_LocalTime"

WMI Persistence

WMI Persistence Example #2: Look for registry key created by using “Win32_LocalTime” from WMI EventFilter:



HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBEM\ESS\//./root/CIMV2\Win32ClockProvider



WDigest (downgrade)

What is the WDigest downgrade?

- Windows 8.1 and Windows 2012 R2 introduced a registry setting that disables storage of clear-text for credentials the WDigest provider. KB2871997 “back-ports” the registry setting to Windows 7, 8, Server 2008R2 and 2012.

Meaning of downgraded WDigest?

- Good indicator to detect potential execution of clear-text credentials extraction like Mimikatz

Location of WDigest indicators?

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential
- If the UseLogonCredential value is set to “1” [opposed to “0”], WDigest will store credentials in cleartext in memory.

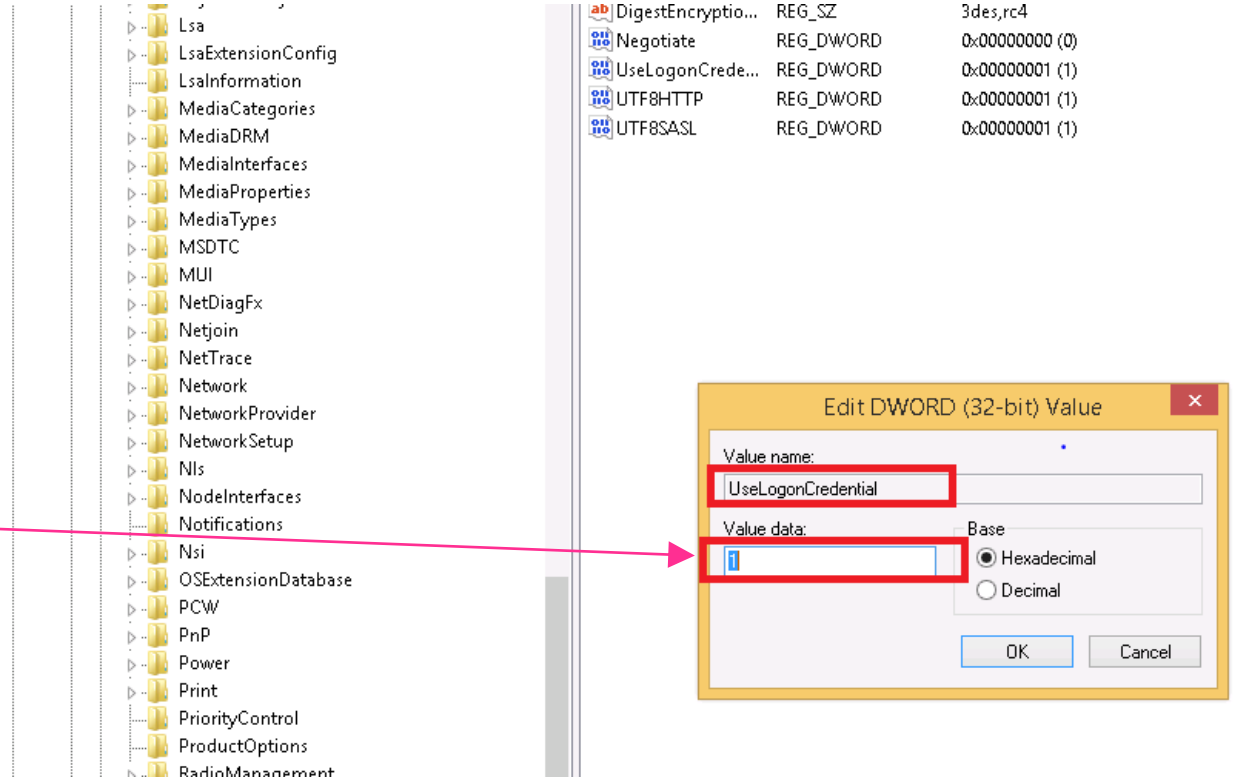
References:

- https://www.trustedsec.com/april-2015/dumping-wdigest-creds-with-meterpreter-mimikatzkiwi-in-windows-8-1
- https://adsecurity.org/?p=559

WDigest downgrade

If UseLogonCredential value is set to 1 in your environment, something may be wrong!

```
reg add
HKLM\SYSTEM\CurrentControlSet\Co
ntrol\SecurityProviders\WDigest
/v UseLogonCredential /t
REG_DWORD /d 1
```



WDigest downgrade

WDigest Example #1: Look for evidence of found modified WDigest values of “1” in tool output:

HostName	WDigestReg Mimikatz Backdoor
10.10.1.16	WDigestReg found:"1"
10.10.1.7	WDigestReg found:"1"
10.10.1.15	WDigestReg found:"1"
10.10.1.6	WDigestReg not found
10.10.1.8	WDigestReg not found
10.10.1.5	WDigestReg not found
10.10.1.10	WDigestReg not found
10.10.1.12	WDigestReg not found

Scheduled Tasks

What are SchTasks?

- A windows feature used to schedule the launch of programs / scripts at pre-defined times
- Tasks can be scheduled via Task Scheduler UI, schtasks.exe, at.exe

Information found in SchTasks?

- Jobs can often contain activities indicating lateral movement and persistence mechanisms

Location of SchTasks indicators?

- Primarily look for “at” *.job creation activities and *.job files created using the “AT” command in C:\Windows\Tasks\ locations:

  C:\Windows\Tasks\At*.job
  C:\Windows\Tasks\Schedlgu.txt

References:

- https://www.blackhat.com/docs/webcast/09172015-leveraging-proactive-defense-rsa.pdf
- https://github.com/sans-dfir/sift-files/blob/master/scripts/jobparse.pl

Scheduled Tasks

Scheduled Tasks Example #1: Look for evidence of malicious "AT" job files and their associated content in tool output:

Suspicious "at" job activities

HostName	Time	Command	Status
10.10.1.6	0	+C:\wmpub\ddumpx64.exe -a >> C:\wmpub\1.txt	Task has not run
10.10.1.5	0	┬C:\wmpub\procdump.bat	Task has not run
10.10.1.5	0	(C:\wmpub\ddump.exe -a >> C:\wmpub\1.txt	Task has not run
10.10.1.5	0	☼C:\wmpub\x.bat	Task has not run
10.10.1.8	0	┬C:\wmpub\procdump.bat	Task has not run
10.10.1.8	0	┬C:\wmpub\procdump.bat	Task has not run
10.10.1.6	0	+C:\wmpub\ddumpx64.exe -a >> C:\wmpub\1.txt	Task has not run

Rogue Services

What are Rogue Services?

- Windows services designed to run programs in the background
- Services can be started as executables or loaded as DLLs without user interaction

Information found in Rogue Services?

- Uncommon services can often be created by attacker to establish persistence mechanisms

How to find Rogue Services?

- Start Type 2 (auto-start) - Service automatically started by the SCM during system startup. Services are started even if a user does not log on.
- Error Control Set to 0- User is not notified if a service fails during startup.
- Random service name - Service name does not look like dictionary word
- Abnormal “ServicePathName” - DLLs loaded by svchost.exe or binaries being run from unusual locations

References:

- https://digital-forensics.sans.org/media/poster_2014_find_evil.pdf
- Intrusion Hunting for the Masses by David Sharpe

Rogue Services

Rogue Services Example #1: Look for evidence of uncommon services in tool output:

Uncommon ServiceCaption

Auto-start & Error Control set to 0

Uncommon service name loaded by svchost

HostName	ServiceName	ServiceCaption	ServiceState	ServiceStartup	ServiceType	ServiceErrorControl	ServicePathName
192.168.100.19	ersvc	error reporting service	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k winerr
192.168.100.19	seclogon	secondary logon	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.19	shellhwdetectio	shell hardware detectio	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.19	vmware physica	vmware physical disk he	running	auto	own process	ignore	c:\program files\vmware\vmware tools\vmacthlp.exe
192.168.100.19	winmgmt	windows management i	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.19	mssecurityupda	security update	running	auto	own process	ignore	c:\windows\system32\svchost.exe -k mssecurityupdat
192.168.100.50	ersvc	error reporting service	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k winerr
192.168.100.50	mssecurityupda	security update	running	auto	own process	ignore	c:\windows\system32\svchost.exe -k mssecurityupdat
192.168.100.50	seclogon	secondary logon	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.50	shellhwdetectio	shell hardware detectio	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.50	vmware physica	vmware physical disk he	running	auto	own process	ignore	c:\program files\vmware\vmware tools\vmacthlp.exe
192.168.100.50	winmgmt	windows management i	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.1	shellhwdetectio	shell hardware detectio	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.1	winmgmt	windows management i	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.6	ersvc	error reporting service	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k winerr
192.168.100.6	seclogon	secondary logon	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs
192.168.100.6	shellhwdetectio	shell hardware detectio	running	auto	share process	ignore	c:\windows\system32\svchost.exe -k netsvcs

Maximizing tool output

- If possible, use a golden image to whitelist data.
- Use a tiered approach to normalize activity: Scan and analyze the tiers of hosts most likely to be the same. For example, separate the server tier from end user workstations.
- Start with a small population to understand how everything works. If you have access, we recommend checking out Domain Controllers.

Maximizing tool output

Efficiencies can be gained with SIEM ingestion

- Automate a portion of the analysis, use pre-existing rules, etc.
- Create tailored rules designed for your corporate environment

Example rule: exclude all the internal destination IP addresses and search for external destination IP addresses against known malicious IP addresses.

Project Details

Download a copy, contribute, or add suggestions:
https://github.com/apthunting/APT-Hunter

Hao Wang: @MrRed_Panda

Joshua Theimer: @6zq